

**THIS PAGE IS INSERTED BY OIPE SCANNING**

**IMAGES WITHIN THIS DOCUMENT ARE BEST AVAILABLE COPY AND CONTAIN DEFECTIVE IMAGES SCANNED FROM ORIGINALS SUBMITTED BY THE APPLICANT.**

**DEFECTIVE IMAGES COULD INCLUDE BUT ARE NOT LIMITED TO:**

**BLACK BORDERS**

**TEXT CUT OFF AT TOP, BOTTOM OR SIDES**

**FADED TEXT**

**ILLEGIBLE TEXT**

**SKEWED/SLANTED IMAGES**

**COLORED PHOTOS**

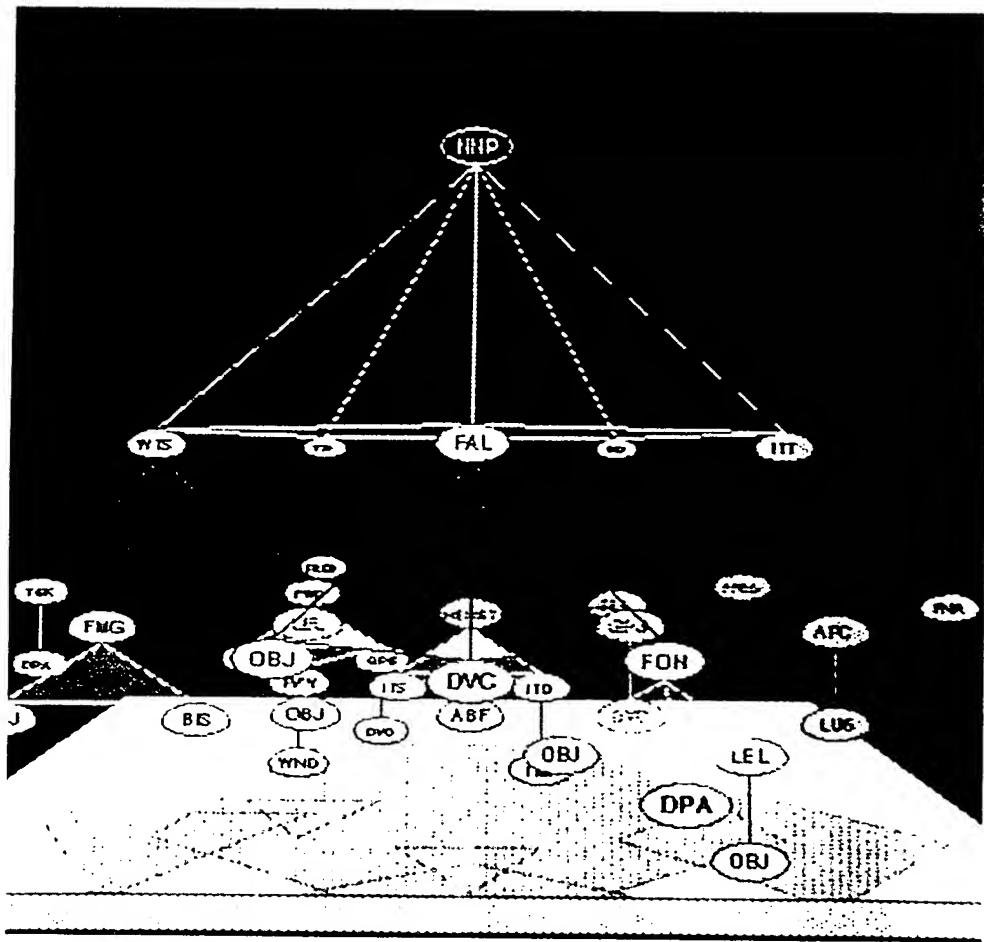
**BLACK OR VERY BLACK AND WHITE DARK PHOTOS** 

**GRAY SCALE DOCUMENTS**

**IMAGES ARE BEST AVAILABLE COPY.  
RESCANNING DOCUMENTS *WILL NOT*  
CORRECT IMAGES.**



# PARASOL



# Object-Oriented and Graphical Database Reference and Visualization

**EXHIBIT B**

# Title: Business Analysis & Management Systems Utilizing Emergent Structures

Inventors: Michael M. Mann & Arne Haugland  
Attns.: Fulwider Patton et al. Dkt. # 65567/ENCPMP

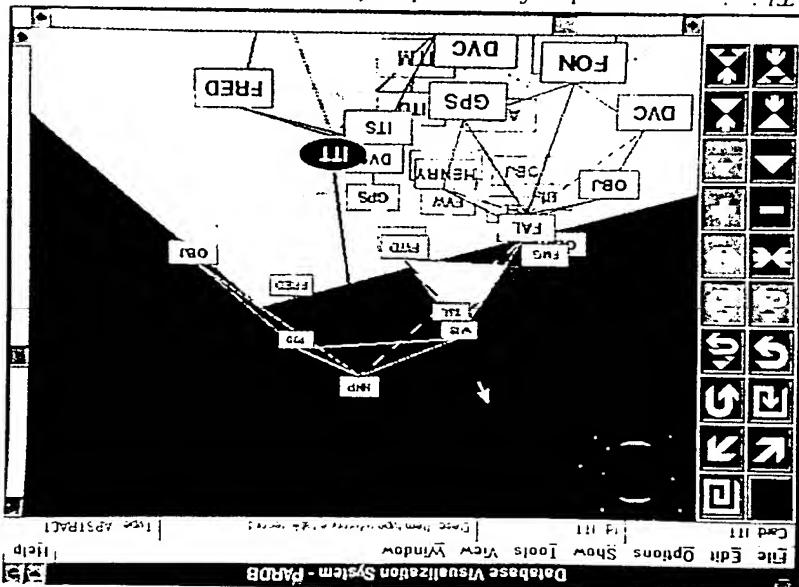
The Parasol product line consists of two related products that provide tools for users and managers to view corporate databases in ways never before possible. Parasol Executive is a ready-to-run library of programs that access SQL-based data tables and present hierarchical structures graphically on the screen. Parasol Development is an application development tool to build custom solutions for entering, reviewing and editing data in SQL tables based on the specific needs of each customer.

PARASOL EXECUTIVE

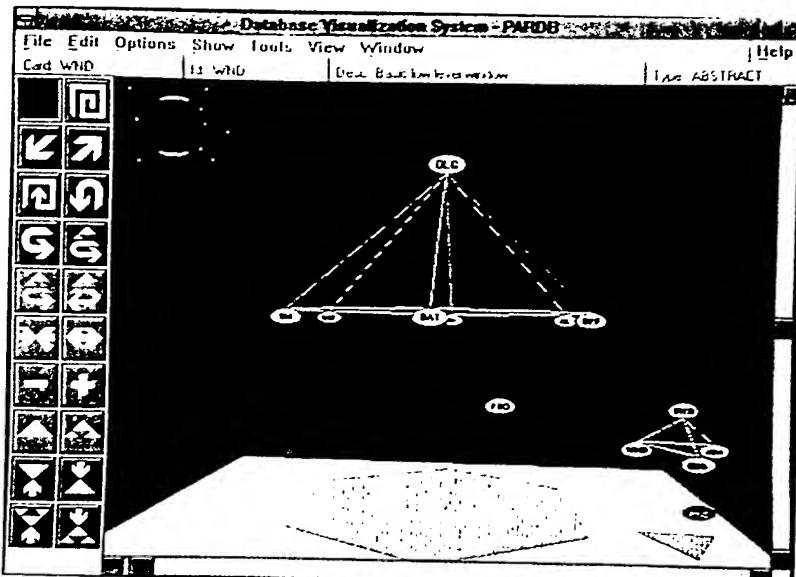
The Parasol product line consists of two related products that provide tools for users and managers to view corporate databases in ways never before possible. Parasol Executive is a ready-to-run library of programs that access SQL-based data tables and present hierarchical structures graphically on the screen. Parasol Developer is an application development tool to build custom solutions for entering, reviewing and editing data in SQL tables based on the specific needs of each customer.

## Parasol Introduction

Parasol can unravel complex, unrelated databases by allowing users to add connections between rows of data in tables. And then graphically viewing them on the screen. Information can be readily accessed, and relationships understood more easily using a graphical view of the data tables. As a result, data tables that may have been considered obsolete can be re-vitalized and used again in making data queries or decisions based on critical information.



This is an example of a couple Parusol Executive, from a 3D



*This screen sample shows the ability of Parasol Executive to remove all data elements except those linked to a selected item.*

Existing databases can be integrated without the need to restructure or transfer information to a new database. This saves time, development effort, and computer operating costs because you can use the existing data, and do not need to convert the large volume of information in your databases.

By using a graphical view of your database, Parasol Executive allows you the ability to find important information without searching through large volumes of reports and data listings. And with its object-oriented design approach, you will be using the most efficient and flexible technologies available for computer software implementation.

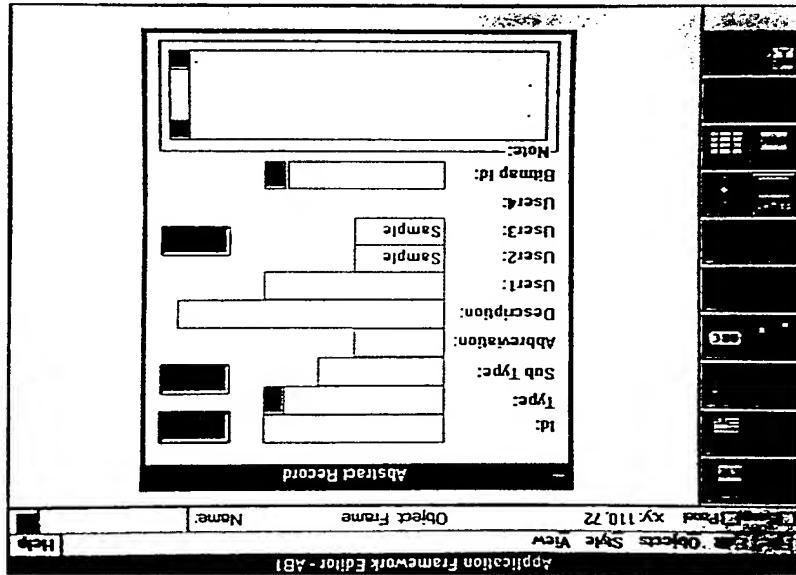
With a 3D view, you can navigate around your database structure, reviewing details as you go. You can look at the structure from any vantage point -- up, down, away, close-up, and even from inside. With this power and flexibility, you will be able to view, analyze and understand data relationships and connections as never before.

Parasol Executive provides standard graphical views, navigational functions, an operational shell, and library modules, to run a custom-built application that you design and generate with Parasol Developer.

This code is then compiled with one easy step, and the custom application is stored ready for execution. With this automatic application generation approach, you save hours of design and development time with a corresponding increase in productivity.

After you identify and specify the tables you require, you design your application with a series of database definitions and dialog box specifications. These are stored as designs and parameters in database form, and then used in the generation of C code for your application.

By using the Alfred (Application Framework Editor) functions of Parallel Desktop, you design and create your own dialog boxes.



Applications that are custom-built and retro-fitted to existing database structures are usually difficult to design, develop, and implement. Parasol Developer provides the tools to design and build object-oriented applications based on un-related SQL tables that may exist on several types of computers and software systems. Using graphical design techniques and an object-oriented approach, you can quickly and easily build your own application with Parasol Developer. And then run it with Parasol Executive.

## Parasol Developer

With Parasol Developer, you create dialog boxes for SQL queries, table views, "where" clauses for filters, and data entry/edit functions. Your final application will have the ability to add connections and links between specific data values of tables that were not previously linked through key fields.

Custom applications created with Parasol Developer have an object-oriented design, and thus can be modified easily. You can change the design specification for any element, re-generate and re-compile the code, and have a new function or feature for your application. Furthermore, you can edit the source code files and add specialized "tweaks" or changes to incorporate other application functionality, or link with programs previously developed. This flexibility and open architecture allows you to easily integrate any Parasol generated application within your corporate information and application environment.

```
!b13    Ab1SetFields( PAb1 pAb1, PGDB pGdb )
{
    PAsr      pAsr;
    LargeInt  offset;
    MediumInt len;

    Ab1Assert( pAb1 );
    GdbAssert( pGdb );

    pAsr = Ab1AsAsr( pAb1 );
    GdbSetText( pGdb, DID_ET_1,      AsrGetAbsid( pAsr ) );
    GdbClearMdt( pGdb, DID_ET_1 );
    GdbSetText( pGdb, DID_COMBO_1,   AsrGetType( pAsr ) );
    GdbClearMdt( pGdb, DID_COMBO_1 );
    GdbSetText( pGdb, DID_ET_2,      AsrGetSubtype( pAsr ) );
    GdbClearMdt( pGdb, DID_ET_2 );
    GdbSetText( pGdb, DID_ET_3,      AsrGetDescription( pAsr ) );
    GdbClearMdt( pGdb, DID_ET_3 );
    GdbSetText( pGdb, DID_ET_4,      AsrGetAttrib( pAsr ) );
    GdbClearMdt( pGdb, DID_ET_4 );
    GdbSetText( pGdb, DID_ET_5,      AsrGetUser1( pAsr ) );
    GdbClearMdt( pGdb, DID_ET_5 );
    GdbSetText( pGdb, DID_ET_6,      AsrGetUser2( pAsr ) );
    GdbClearMdt( pGdb, DID_ET_6 );
}
Line=652 Col=1          A:AB1.C      Insert  WA-632221.indd
```

*The source code generated by Parasol Developer is object-oriented in design, and can be edited for further customization or refinement.*

By using Parasol's unique ability to access existing SQL databases without restructuring, you can create graphical database reference and visualization applications within hours and days, rather than weeks and months. Also, with generated code, less QA and testing time is required, since the source code files are created by computer software rather than manually typed by programmers.

## Features of Parasol Executive

- ### View of Connected Tables

  - > Data links can be defined between rows of data in the same table or in other tables of the database
  - > Based on the links established by the user, the data tables are shown in a 3D graphic or 2D outline view of the database structure
  - > Links can be established between rows of data, whether or not foreign key fields were previously defined for linking those specific tables
  - > You can condense large, complex data structures to smaller, more manageable views with filtering and other selection functions
  - > Long lists and reports can be simplified to graphical pictures, both on screen and in hardcopy reports
- ### Reduced Data View

  - > Data links can be established between rows of data, whether or not foreign key fields were previously defined for linking those specific tables
  - > You can use around the structure view by pointing at the element you want to see, or by moving the view with a position indicator
  - > When you select an item, you can view all the details about it, and edit the data
  - > Data element connections can be added and changed with point-and-click methods
  - > With a graphical view of the database, you can visualize the impact of various changes to the data, or conditions for the various data elements
  - > You can see what item connects to can be simplified and analyzed easily
  - > Information and what it connects to can be simplified and analyzed easily
  - > You can use an inverted view of the data structure to evaluate information from a "bottoms-up" approach to understand where an item is used
  - > You can use an inverted view of the data structure to evaluate information from a "tops-down" approach to understand what items are used
- ### Impact Analysis

  - > You can move around the structure view by pointing at the element you want to see, or by moving the view with a position indicator
  - > When you select an item, you can view all the details about it, and edit the data
  - > Data element connections can be added and changed with point-and-click methods
  - > With a graphical view of the database, you can visualize the impact of various changes to the data, or conditions for the various data elements
  - > You can see what item connects to can be simplified and analyzed easily
  - > Information and what it connects to can be simplified and analyzed easily
  - > You can use an inverted view of the data structure to evaluate information from a "bottoms-up" approach to understand where an item is used
  - > You can use an inverted view of the data structure to evaluate information from a "tops-down" approach to understand what items are used
- ### Data Implosion

  - > You can use an inverted view of the data structure to evaluate information from a "bottoms-up" approach to understand where an item is used
  - > You can use an inverted view of the data structure to evaluate information from a "tops-down" approach to understand what items are used
  - > You can use an inverted view of the data structure to evaluate information from a "bottoms-up" approach to understand where an item is used
  - > You can use an inverted view of the data structure to evaluate information from a "tops-down" approach to understand what items are used
- ### Data Update and Reference

  - > Any data values in the tables can be added, changed or reviewed
  - > You can integrate text, graphics and application programs associated with various data elements in the structure
  - > Links can be added, changed or deleted
  - > Elements in the structure
- ### Multiple Data Links

  - > Any group of tables can be linked on a one-to-one, one-to-many, or many-to-many basis
  - > You connect rows in tables to another row in the same table or other tables, whether
  - > All information can be viewed as a 2D outline of text
  - > Custom reports and tabular listings can be generated and printed
- ### Outline Processing

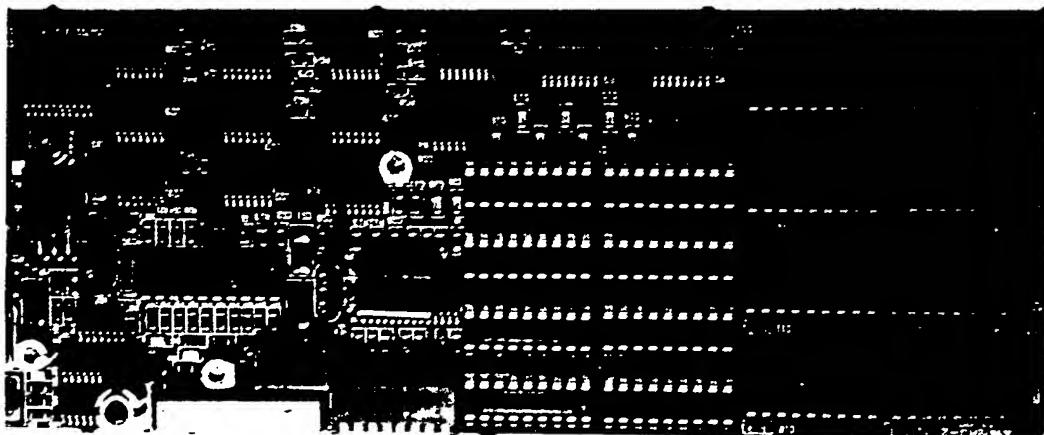
## Other Parasol Products

Other products available from Parasol Inc. include sample applications for various industries and company functions, and an electronic form of the Parasol Executive User Guide that can be customized by each client for their own specific application. Contact Parasol Inc. for further information about pricing and availability of these Parasol products.

## System Requirements

The following computer system components are required for the Parasol products:

IBM OS/2 V1.3 or later  
IBM Database Manager for OS/2  
386/20 or higher CPU with 8 MB memory  
Floating Point Processor (for 386 based machines)  
VGA graphics card and monitor (XGA preferable)  
SQL database access to application datatables (if used in the application design)  
For Parasol Executive: 5 MB of disk space for each application  
For Parasol Developer: 20 MB of disk space  
C compiler (IBM or Microsoft)  
OS/2 Software Development Kit (SDK)



**PARASOL, INC.**

2551 Pebble Beach Loop, Lafayette, California 94549  
Phone: 510-934-8768 Fax: 510-934-8112

Parasol is a trademark of Parasol Inc. Other trademarks are properties of their respective companies.  
© Copyright 1992 Parasol Inc.

### *Time Based Relationships*

- > You can attach effective dates to any data element, and use these for selection criteria in filters (before and after type queries)
- > This function allows you to manage change to the database or structure

### *Change Management*

- > Automatic tracking of database changes including, audit trails and other information, makes database maintenance easy and simple

### *Data Table Filtering*

- > You use standard SQL query statements to extract data from the database, based on only the information you want to view
- > Various views are stored and accessible directly from the main menus once they are generated
- > Enter filters on an individual table basis

## **Features of Parasol Developer**

### *Standard Naming Conventions*

- > Data files, code, and objects utilize standard naming conventions that you can use
- > Data tables are defined, either as new tables in Database Manager, or as existing tables in other SQL databases

### *Customize Menu Structure*

- > Based on the unique data tables for each application, you can add and/or change the menu names for each user of Parasol Executive
- > You can edit other C source code files provided with Parasol Developer to make changes to how Parasol Executive will behave

### *Design the Application*

- > Use the Alfred (Application Framework Editor) functions to design and build all record queries, table views, find dialogs and data dialogs for the application
- > Point-and-click methods and standard graphical icons are used to create object-oriented applications customized for each user

### *Code Generation*

- > Run the Parasol Code Generator to create object-oriented C source code files for each object in the application, and other source code files provided
- > Compile the source code into executable libraries and functions

### *Make the Application*

- > You can edit any of the source code files to add extra functions, or integrate other applications or modules
- > Link the entire application together and merge it into the library modules of Parasol Executive with one execution file
- > Install an icon in OS/2 and attach the application database and executable file for a user to start the customized application